# 2014 Latest Oracle 1Z0-051 Exam Demo Free Download!

QUESTION 1 Evaluate the following CREATE TABLE commands: CREATE TABLE orders (ord_no NUMBER(2) CONSTRAINT ord_pk PRIMARY KEY, ord_date DATE, cust_id NUMBER(4)); CREATE TABLE ord_items (ord_no NUMBER(2), item_no NUMBER(3), qty NUMBER(3) CHECK (qty BETWEEN 100 AND 200), expiry_date date CHECK (expiry_date > SYSDATE), CONSTRAINT it_pk PRIMARY KEY (ord_no,item_no), CONSTRAINT ord_fk FOREIGN KEY(ord_no) REFERENCES orders(ord_no)); The above command fails when executed. What could be the reason? A. SYSDATE cannot be used with the CHECK constraint. B.    The BETWEEN clause cannot be used for the CHECK constraint. C. The CHECK constraint cannot be placed on columns having the DATE data type. D.    ORD_NO and ITEM_NO cannot be used as a composite primary key because ORD_NO is also the FOREIGN KEY. Answer: A Explanation: CHECK Constraint The CHECK constraint defines a condition that each row must satisfy. The condition can use the same constructs as the query conditions, with the following exceptions: References to the CURRVAL, NEXTVAL, LEVEL, and ROWNUM pseudocolumns Calls to SYSDATE, UID, USER, and USERENV functions Queries that refer to other values in other rows A single column can have multiple CHECK constraints that refer to the column in its definition. There is no limit to the number of CHECK constraints that you can define on a column. CHECK constraints can be defined at the column level or table level. CREATE TABLE employees (... salary NUMBER(8,2) CONSTRAINT emp_salary_min CHECK (salary > 0), QUESTION 2 View the Exhibit and examine the structure of the CUSTOMERS table. Using the CUSTOMERS table, y ou need to generate a report that shows an increase in the credit limit by 15% for all customers. Customers whose credit limit has not been entered should have the message " Not Available" displayed. Which SQL statement would produce the required result?

| Table CUSTOMERS | | |
|---|---|---|
| Name | Null? | Type |
| CUST_ID | NOT NULL | NUMBER |
| CUST_FIRST_NAME | NOT NULL | VARCHAR2 (20) |
| CUST_LAST_NAME | NOT NULL | VARCHAR2 (40) |
| CUST_GENDER | NOT NULL | CHAR (1) |
| CUST_YEAR_OF_BIRTH | NOT NULL | NUMBER (4) |
| CUST_MARITAL_STATUS | | VARCHAR2 (2) |
| CUST_STREET_ADDRESS | NOT NULL | VARCHAR2 (40) |
| CUST_POSTAL_CODE | NOT NULL | VARCHAR2 (10) |
| CUST_CITY | NOT NULL | VARCHAR2 (30) |
| CUST_STATE_PROVINCE | NOT NULL | VARCHAR2 (40) |
| COUNTRY_ID | NOT NULL | NUMBER |
| CUST_INCOME_LEVEL | | VARCHAR2 (30) |
| CUST_CREDIT_LIMIT | | NUMBER |
| CUST_EMAIL | | VARCHAR2 (30) |

A.    SELECT NVL(cust_credit_limit,'Not Available')*.15 "NEW CREDIT" FROM customers; B.    SELECT NVL(cust_credit_limit*.15,'Not Available') "NEW CREDIT" FROM customers; C.    SELECT TO_CHAR(NVL(cust_credit_limit*.15,'Not Available')) "NEW CREDIT" FROM customers; D.    SELECT NVL(TO_CHAR(cust_credit_limit*.15),'Not Available') "NEW CREDIT" FROM customers; Answer: D Explanation: NVL Function Converts a null value to an actual value: Data types that can be used are date, character, and number. Data types must match: NVL(commission_pct,0) NVL(hire_date,'01-JAN-97') NVL(job_id,'No Job Yet') QUESTION 3 Examine the structure of the PROGRAMS table: name        Null        Type ------        ---------    ------- PROG_ID        NOT NULL    NUMBER(3) PROG_COST            NUMBER(8,2) START_DATE    NOT NULL    DATE END_DATE DATE Which two SQL statements would execute successfully? (Choose two.) A.    SELECT NVL(ADD_MONTHS(END_DATE,1),SYSDATE) FROM programs; B.    SELECT TO_DATE(NVL(SYSDATE-END_DATE,SYSDATE)) FROM programs; C.    SELECT NVL(MONTHS_BETWEEN(start_date,end_date),'Ongoing') FROM programs; D.    SELECT NVL(TO_CHAR(MONTHS_BETWEEN(start_date,end_date)),'Ongoing') FROM programs; Answer: AD Explanation: NVL Function Converts a null value to an actual value: Data types that can be used are date, character, and number. Data types must match: NVL(commission_pct,0) NVL(hire_date,'01-JAN-97') NVL(job_id,'No Job Yet') MONTHS_BETWEEN(date1, date2): Finds the number of months between date1 and date2 The result can be positive or negative. If date1 is later than date2, the result is positive; if date1 is earlier than date2, the result is negative. The noninteger part of the result represents a portion of the month. MONTHS_BETWEEN returns a numeric value. - answer C NVL has different datatypes - numeric and strings, which is not possible! The data types of the original and if null parameters must always be compatible. They must either be of the same type, or it must be possible to implicitly convert if null to the type of the original parameter. The NVL function returns a value with the same data type as the original parameter. QUESTION 4 The PRODUCTS table has the following structure: name        Null Type ------        ---------    ------- PROD_ID        NOT NULL    NUMBER(4) PROD_NAME VARCHAR2(25) PROD_EXPIRY_DATE        DATE Evaluate the following two SQL statements: SQL>SELECT prod_id, NVL2(prod_expiry_date, prod_expiry_date + 15,'') FROM products; SQL>SELECT prod_id, NVL(prod_expiry_date,

prod_expiry_date + 15) FROM products; Which statement is true regarding the outcome? A.    Both the statements execute and give different results. B.    Both the statements execute and give the same result. C.    Only the first SQL statement executes successfully. D.    Only the second SQL statement executes successfully. Answer: A Explanation: Using the NVL2 Function The NVL2 function examines the first expression. If the first expression is not null, the NVL2 function returns the second expression. If the first expression is null, the third expression is returned. Syntax NVL2(expr1, expr2, expr3) In the syntax: expr1 is the source value or expression that may contain a null expr2 is the value that is returned if expr1 is not null expr3 is the value that is returned if expr1 is null QUESTION 5 Examine the structure of the INVOICE table. Name        Null        Type  ------    ---------    ------- INV_NO    NOT NULL    NUMBER(3) INV_DATE            DATE INV_AMT            NUMBER(10,2) Which two SQL statements would execute successfully? (Choose two.) A.    SELECT inv_no,NVL2(inv_date,'Pending','Incomplete') FROM invoice; B.    SELECT inv_no,NVL2(inv_amt,inv_date,'Not Available') FROM invoice; C.    SELECT inv_no,NVL2(inv_date,sysdate-inv_date,sysdate) FROM invoice; D.    SELECT inv_no,NVL2(inv_amt,inv_amt*.25,'Not Available') FROM invoice; Answer: AC Explanation: The NVL2 Function The NVL2 function provides an enhancement to NVL but serves a very similar purpose. It evaluates whether a column or expression of any data type is null or not. 5-6 The NVL function If the first term is not null, the second parameter is returned, else the third parameter is returned. Recall that the NVL function is different since it returns the original term if it is not null. The NVL2 function takes three mandatory parameters. Its syntax is NVL2(original, ifnotnull, ifnull), where original represents the term being tested. Ifnotnull is returned if original is not null, and ifnull is returned if original is null. The data types of the ifnotnull and ifnull parameters must be compatible, and they cannot be of type LONG. They must either be of the same type, or it must be possible to convert ifnull to the type of the ifnotnull parameter. The data type returned by the NVL2 function is the same as that of the ifnotnull parameter. QUESTION 6 Evaluate the following CREATE TABLE commands: CREATE TABLE orders (ord_no NUMBER(2) CONSTRAINT ord_pk PRIMARY KEY, ord_date DATE, cust_id NUMBER(4)); CREATE TABLE ord_items (ord_no NUMBER(2), item_no NUMBER(3), qty NUMBER(3) CHECK (qty BETWEEN 100 AND 200), expiry_date date CHECK (expiry_date > SYSDATE), CONSTRAINT it_pk PRIMARY KEY (ord_no,item_no), CONSTRAINT ord_fk FOREIGN KEY(ord_no) REFERENCES orders(ord_no)); The above command fails when executed. What could be the reason? A.    SYSDATE cannot be used with the CHECK constraint. B.    The BETWEEN clause cannot be used for the CHECK constraint. C.    The CHECK constraint cannot be placed on columns having the DATE data type. D.    ORD_NO and ITEM_NO cannot be used as a composite primary key because ORD_NO is also the FOREIGN KEY. Answer: A QUESTION 7 Which two statements about sub queries are true? (Choose two.) A.    A sub query should retrieve only one row. B.    A sub query can retrieve zero or more rows. C.    A sub query can be used only in SQL query statements. D.    Sub queries CANNOT be nested by more than two levels. E.    A sub query CANNOT be used in an SQL query statement that uses group functions. F.    When a sub query is used with an inequality comparison operator in the outer SQL statement, the column list in the SELECT clause of the sub query should contain only one column. Answer: BF Explanation: sub query can retrieve zero or more rows, sub query is used with an inequality comparison operator in the outer SQL statement, and the column list in the SELECT clause of the sub query should contain only one column. Incorrect answer: Asub query can retrieve zero or more rows Csub query is not SQL query statement Dsub query can be nested Egroup function can be use with sub query QUESTION 8 Which three statements are true regarding subqueries? (Choose three.) A.    Subqueries can contain GROUP BY and ORDER BY clauses. B.    Main query and subquery can get data from different tables. C.    Main query and subquery must get data from the same tables. D.    Subqueries can contain ORDER BY but not the GROUP BY clause. E.    Only one column or expression can be compared between the main query and subquery. F.    Multiple columns or expressions can be compared between the main query and subquery. Answer: ABF Explanation: SUBQUERIES can be used in the SELECT list and in the FROM, WHERE, and HAVING clauses of a query. A subquery can have any of the usual clauses for selection and projection. The following are required clauses: A SELECT list A FROM clause The following are optional clauses: WHERE GROUP BY HAVING The subquery (or subqueries) within a statement must be executed before the parent query that calls it, in order that the results of the subquery can be passed to the parent. QUESTION 9 Which statement is true regarding the UNION operator? A.    The number of columns selected in all SELECT statements need to be the same B.    Names of all columns must be identical across all SELECT statements C.    By default, the output is not sorted D.    NULL values are not ignored during duplicate checking Answer: A Explanation: The SQL UNION query allows you to combine the result sets of two or more SQL SELECT statements. It removes duplicate rows between the various SELECT statements. Each SQL SELECT statement within the UNION query must have the same number of fields in the result sets with similar data types. QUESTION 10 Examine the structure of the PROGRAMS table: name        Null        Type  ------    ---------    ------- PROG_ID    NOT NULL    NUMBER(3) PROG_COST            NUMBER(8,2) START_DATE    NOT NULL    DATE END_DATE DATE Which two SQL statements would execute successfully? (Choose two.) A.    SELECT

NVL(ADD_MONTHS(END_DATE,1),SYSDATE) FROM programs; B.   SELECT TO_DATE(NVL(SYSDATE-END_DATE,SYSDATE)) FROM programs; C.   SELECT NVL(MONTHS_BETWEEN(start_date,end_date),'Ongoing') FROM programs; D.   SELECT NVL(TO_CHAR(MONTHS_BETWEEN(start_date,end_date)),'Ongoing') FROM programs; Answer: AD QUESTION 11 The PRODUCTS table has the following structure: name              Null      Type   ------         ---------   ------- PROD_ID NOT NULL    NUMBER(4) PROD_NAME                        VARCHAR2(25) PROD_EXPIRY_DATE          DATE Evaluate the following two SQL statements: SQL>SELECT prod_id, NVL2(prod_expiry_date, prod_expiry_date + 15,") FROM products; SQL>SELECT prod_id, NVL(prod_expiry_date, prod_expiry_date + 15) FROM products; Which statement is true regarding the outcome? A.   Both the statements execute and give different results. B.   Both the statements execute and give the same result. C. Only the first SQL statement executes successfully. D.   Only the second SQL statement executes successfully. Answer: A QUESTION 12 Evaluate the following SQL statement: SQL> SELECT promo_id, promo_category FROM promotions WHERE promo_category = 'Internet' ORDER BY 2 DESC UNION SELECT promo_id, promo_category FROM promotions WHERE promo_category = 'TV' UNION SELECT promo_id, promo_category FROM promotions WHERE promo_category ='Radio'; Which statement is true regarding the outcome of the above query? A.   It executes successfully and displays rows in the descending order of PROMO_CATEGORY. B.   It produces an error because positional notation cannot be used in the ORDER BY clause with SET operators. C.   It executes successfully but ignores the ORDER BY clause because it is not located at the end of the compound statement. D.   It produces an error because the ORDER BY clause should appear only at the end of a compound query-that is, with the last SELECT statement. Answer: D Explanation: Using the ORDER BY Clause in Set Operations The ORDER BY clause can appear only once at the end of the compound query. Component queries cannot have individual ORDER BY clauses. The ORDER BY clause recognizes only the columns of the first SELECT query. By default, the first column of the first SELECT query is used to sort the output in an ascending order. Passing your Oracle 1Z0-051 Exam by using the latest Oracle 1Z0-051 Exam Demo Full Version: http://www.braindump2go.com/1z0-051.html